

Erasure Information for a Reed-Solomon Decoder

G. H. Pitt, III, and L. Swanson
Communications Systems Research Section

Many Reed-Solomon decoders, including the one decoding the outer code for Voyager data from Uranus, assume that all symbols have the same chance of being correct or incorrect. In some cases, like in a burst of incorrect symbols, this is not the case, and a Reed-Solomon decoder could make use of this. We examine the use of information about bit quality sent to the Reed-Solomon from an (inner) Viterbi decoder and information about the error status of adjacent symbols in decoding interleaved Reed-Solomon encoded symbols, and discover that, in a region of interest, only about 0.04 dB can be gained.

I. Introduction

A digital coding system, used on Voyager and as an international coding standard, is shown in Fig. 1. The decoding for this system, as implemented in the deep space network and at JPL, is shown in Fig. 2. The Reed-Solomon code is able to make use of soft-quantized data, however, especially in the form of symbol erasures. This article considers the value of passing "erasure information" to the Reed-Solomon decoder.

Our Reed-Solomon code is an 8-bit (255, 223) code. This means that each word consists of 255 symbols of 8 bits each. Of these symbols, 223 are information and 32 are parity. The code allows any 16 *symbol* errors to be corrected. But the code in fact has greater erasure correction capability. If some symbols are lost, or if there is reason to believe that some symbols are in error, they can be declared "erasures." The code can correct any word in which $2e + E \leq 32$, where e is the number of errors in the word and E is the number of erasures.

From time to time, methods have been discussed to make use of this erasure correction capacity. One is to develop a

method of determining quality information for Viterbi decoded bits from the rate of decoder metric renormalization, and to erase those symbols which contain bits of bad quality. Another is to make use of the interleaving of Reed-Solomon words (Fig. 3) to erase symbols which are adjacent to incorrect symbols. (This uses the fact that errors in a Viterbi decoded stream fall in "bursts"; Ref. 1.) We have used a software Viterbi decoder simulator written by Fabrizio Pollara to simulate a number of possible methods for the Viterbi decoder to pass quality information bits; these are described below, but none of them is satisfactory. We have studied the method of using information from interleaved words; this gains only about 0.04 dB, but at some signal-to-noise ratios this improves the bit error rate by about 50%. (This method may be used by the European Space Agency during its Giotto mission.)

II. Erasure Information from the Viterbi Decoder

A problem with determining quality information for Viterbi decoded bits from the rate of metric renormalization is

that we don't know how the metric renormalization is implemented in the Viterbi decoder, but we can simulate it by examining the change in the best (smallest) metric over time. We assumed that we could find the exact best metric after any bit and compare the observed byte errors to the difference in the best metric over 8 consecutive bits. This method should give us results at least as good as in any kind of renormalization scheme.

It is unclear whether the highest correlation will occur by comparing the difference in metrics from the beginning of a byte to the end of it, or from the n th bit in the byte to the n th in the next, or even by comparing one byte's metrics to another byte, so we used programs that would allow the user to try all of these schemes (Fig. 4). The difference in best metric (BM) was computed from the n th bit in one byte to the n th bit in the next and compared to the m th byte away. Then m and n were varied to achieve the highest correlation.

After shifting bits and metrics to find the highest correlation on several simulations, we found very little correlation at all. We used a program that assumes that all bytes with a corresponding metric above a certain threshold (set by the user) will be erased, and calculates the improvement gained in bytes. For example, an improvement of 2 bytes means that the Reed-Solomon decoder will react (decode or not) as though there were 2 fewer byte errors than before. Of course, this number means nothing unless compared with the number of bytes observed.

With 896 bytes observed at 0 dB, the best gain found was 15.5 bytes over the 373 bytes in error using conventional means, a gain in byte error rate from 0.364 to 0.349. Reed-Solomon frames essentially never decode at either of these error rates (less than 0.001% of the time), so there is no gain. There can be gain from erasures only if the Reed-Solomon word decodes after erasures.

With 3968 bytes observed at 1.5 dB (an interesting area for Reed-Solomon encoded data), the best gain found was 1 byte (2 predicted errors and 2 occurrences) over the conventional 201 bytes in error, a gain in byte error rate from 0.0507 to 0.0504. Considering the fact that there would almost never be more than one erasure in a frame (less than 1% of the time), this is no real gain at all. There can be gain from erasures only if there are at least two of them.

III. Erasure Information From Interleaved Reed-Solomon Words

Examining erasure based on interleaving of Reed-Solomon codewords proved more profitable. We ran the simulation for a long time and made a graph of the symbol error bursts (Fig.

5). Then, given an undecoded word W_i , we found the probability that the previous word W_{i-1} decoded and erased all symbols in W_i following those symbols which were corrected in W_{i-1} . Possibly an even better scheme is erasing only symbols in W_i surrounded by two detected in error (in W_{i-1} and W_{i+1}), which works only if the words on both sides of W_i decode.

Taking into account the fact that not all undecoded words have an adjacent decoded word (only 77% do at 1.7 dB), and even given that an adjacent word decodes the probability of decoding is not 100% (37% will then decode), only 29% of the previously undecoded words will decode with this scheme (Fig. 6).

The gain may now be calculated by making a graph of the probability of an undecoded word for the regular decoder and for the new one (Fig. 7). The calculated estimate of the probability of decoding previously undecoded words for 2 dB was about the same as for 1.7 dB, so the gain for this one-sided scheme is about 0.02 dB.

At 1.7 dB, the statistics in Fig. 5 show that if the symbol before a given symbol is in error but the one after it is not, the symbol in between is more likely to be correct than incorrect. Therefore, if an undecoded word is surrounded by two decoded ones, erasing all symbols which are adjacent to exactly one detected in error loses more than it gains. However, if only symbols surrounded on both sides by symbols are erased, 10 times as many symbols in error as correct ones will be erased.

We calculated that 61% of the undecoded words are surrounded by two decoded words. Even those 61% will not decode with probability 1, but a good guess (derived from the assumptions in the appendix) is that about 81% will now decode. Therefore, about 50% of the previously undecoded words will now decode at 1.7 dB (Fig. 6).

The gain may be calculated by making a graph of the probability of an undecoded word for the regular decoder and for the new one (Fig. 7). We assume that the estimated probability of decoding previously undecoded words for 2 dB is about the same as for 1.7 dB, so the curve may be drawn. The gain for this two-sided scheme is about 0.03 dB.

From the above information, we may naively estimate the probability of decoding previously undecoded words if we try the two-sided scheme when both W_{i-1} and W_{i+1} decode and the one-sided scheme if only one of them decodes.

The probability that exactly one side decodes is $2 \times (77\% - 61\%) = 32\%$ and 29% of those will now decode, so we get $50\% + (32\% \times 29\%) = 59\%$ of the previously undecoded words to now decode (Fig. 6).

The gain may be calculated by making a graph of the probability of an undecoded word for the regular decoder and for the new one (Fig. 7). We assume that the estimated probability of decoding previously undecoded words for 2 dB is about the same as for 1.7 dB, so the curve may be drawn. The gain for this combination of the two schemes is about 0.04 dB.

Figures 6 and 7 show the probability of not decoding a word instead of the probability of not decoding a single

symbol. To find that, one must multiply by the fraction of symbols in error given that a word doesn't decode (the mean number of symbols in error given that a word doesn't decode divided by 255). These new schemes decode most of the undecoded words with a low number of errors, so the fraction of errors given that a word doesn't decode will increase and the symbol error rate gain may not be quite as good. (The change in mean number of symbols in error given that a word doesn't decode should be about from 20 to 21.)

Reference

1. Miller, R. L. , Deutsch, L. J., and Butman, S. A., *On the Error Statistics of Viterbi Decoding and the Performance of Concatenated Codes*, JPL Publication 81-9, Jet Propulsion Laboratory, Pasadena, Calif., September 1, 1981.

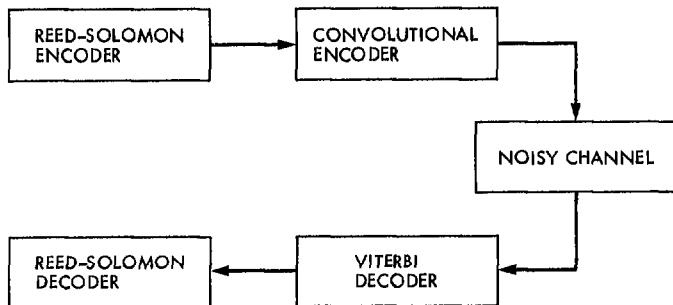


Fig. 1. A digital data coding scheme

WORD 1	WORD 2	WORD 3	WORD 4	WORD 5
SYMBOL 1	SYMBOL 2	SYMBOL 3	SYMBOL 4	SYMBOL 5
SYMBOL 6	SYMBOL 7	SYMBOL 8	SYMBOL 9	SYMBOL 10
SYMBOL 11	SYMBOL 12	SYMBOL 13	SYMBOL 14	SYMBOL 15
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
SYMBOL 1271	SYMBOL 1272	SYMBOL 1273	SYMBOL 1274	SYMBOL 1275

Fig. 3. The 5 interleaved (vertical) Reed-Solomon codewords of 255 symbols each. Notice the sequence of the symbols.

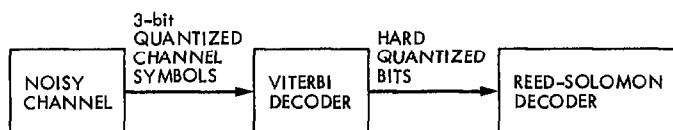


Fig. 2. The current decoding method

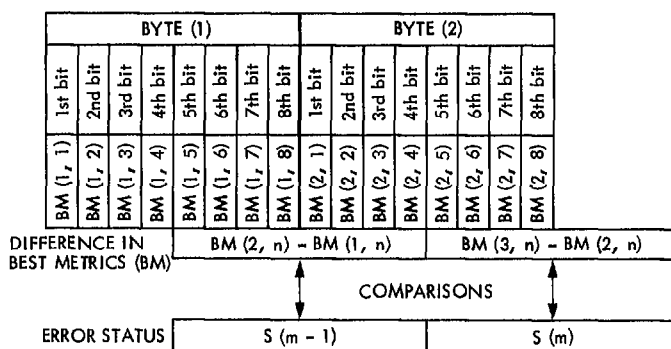


Fig. 4. Difference in best metric (BM) vs byte error status (S) comparisons. Variable n is the bit in the byte where you want to start computing and m is the byte shift.

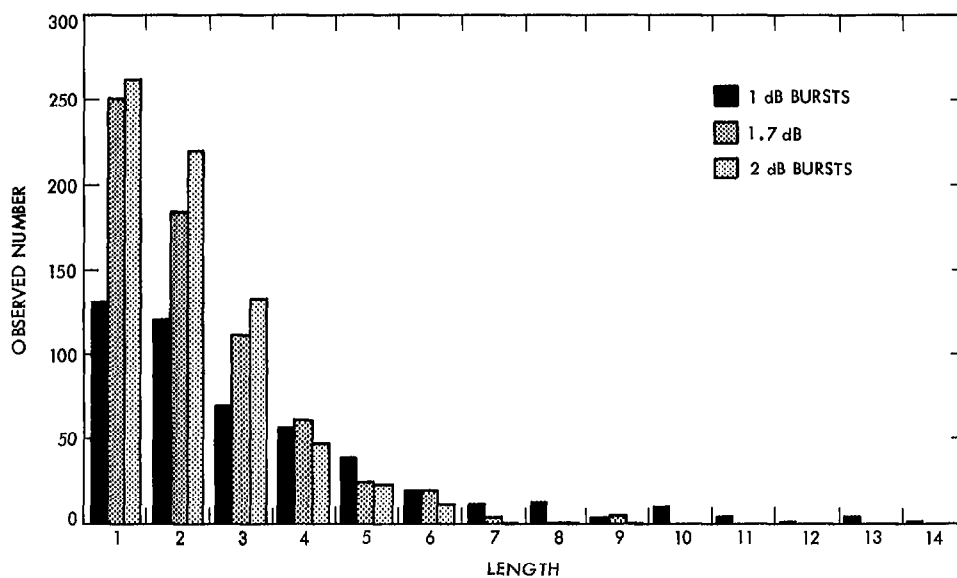


Fig. 5. A graph of the length of bursts for 1, 1.7, and 2 dB as observed in simulation and normalized to 1500 errors at each E_b/N_0 .

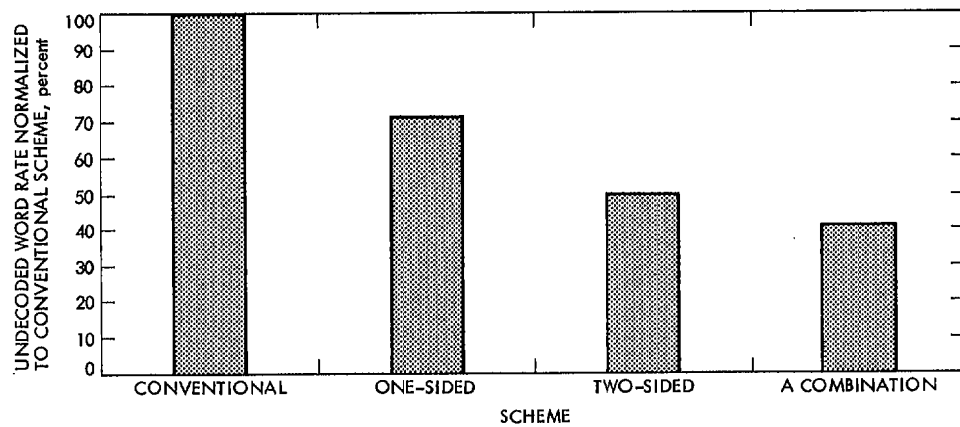


Fig. 6. A graph comparing decoded word rates for the old scheme and the three new ones

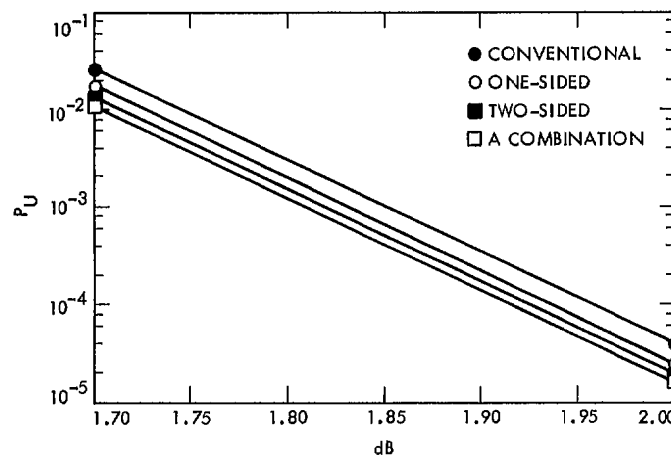


Fig. 7. A graph of the probability of an undecoded word for the old scheme and the three new ones

Appendix

The above analyses may all be derived from the following assumptions (Viterbi decoder errors occur in “bursts”; Ref. 1) and observations using the software Viterbi decoder at 1.7 dB.

Total number of symbols = 38528

Total number of errors = 1497

Total number of bursts = 659

Number of correct symbols with two adjacent incorrect symbols = 33

Total number of triple errors = 430

If W_i = word i and E_i = the number of errors in W_i then we assume:

$P(W_i \text{ now decodes} \mid W_{i-1}, W_{i+1} \text{ decoded and } E_i = 17) = 100\%$

$P(W_i \text{ now decodes} \mid W_{i-1}, W_{i+1} \text{ decoded and } E_i = 18) = 90\%$

$P(W_i \text{ now decodes} \mid W_{i-1}, W_{i+1} \text{ decoded and } E_i = 19) = 55\%$

$P(W_i \text{ now decodes} \mid W_{i-1}, W_{i+1} \text{ decoded and } E_i = 20) = 10\%$

$P(W_i \text{ now decodes} \mid W_{i-1}, W_{i+1} \text{ decoded and } E_i = 21 \text{ to } 255) = 0\%$

In the one-sided scheme, we assumed that only symbols that were adjacent to a symbol that decoded with the normal decoder may now be affected by the new new scheme. This makes the estimate a little low since we could decode one word with the new scheme and then use that information to decode the word next to it, but this would seem to happen so infrequently that we didn't try to estimate its probability.